Bielefeld University
Faculty of Technology

AG Genominformatik
Prof. Dr. Jens Stoye
Dr. Pedro Feijão

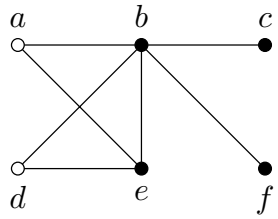
Algorithms in Genome Research

Winter 2015/2016

Exercises

Number 9, Discussion: 2016 January 22

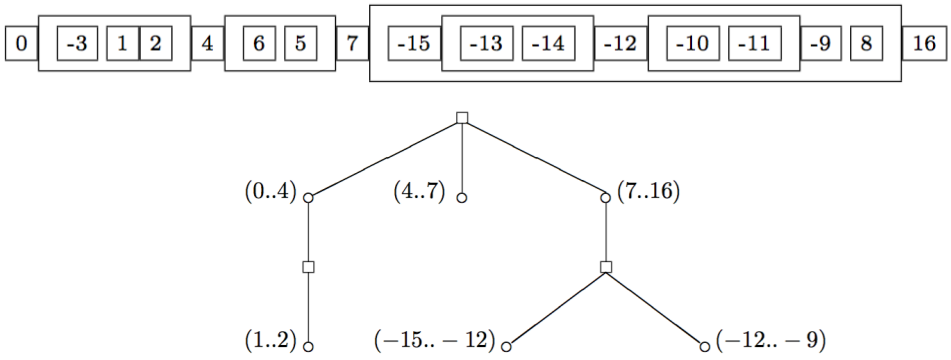
1. Suppose that a given permutation has the following overlap graph:



   (a) What is the oriented vertex with maximum score? Apply the reversal defined by this vertex, update the overlap graph, and repeat the process until the permutation is sorted.

   (b) Can you find a breakpoint graph and a permutation that correspond to the overlap graph in the figure?

2. Find the components and the component tree for the following permutations. For instance, for $\pi = (0\ -3\ 1\ 2\ 4\ 6\ 5\ 7\ -15\ -13\ -14\ -12\ -10\ -11\ -9\ 8\ 16)$ we have



   (a) $\pi = (\ 0\ 3\ 10\ 9\ 4\ 2\ 1\ 5\ 7\ 6\ 8\ 11\ 13\ 12\ 14\ )$.

   (b) $\pi = (\ 0\ 1\ 3\ -5\ 4\ 6\ 2\ 7\ )$

   (c) $\pi = (\ 0\ 2\ 4\ 3\ 5\ 1\ 6\ 8\ 7\ 10\ )$

3. An *Unoriented component* is a component that has no oriented pairs. Oriented pairs in *nested components* (inside of a component) do not count for the main component (This happens in example (b) of the previous exercise, for instance). It is also possible to detect unoriented components on the overlap graph,

(a) How many unoriented components there are in the permutations of Exercise 2?

(b) The reversal distance is $d = n - c + k$, where k represents the extra cost of sorting unoriented components, and $k = 0$ if there are no unoriented components. In the lecture, we found examples where k is exacly the number of unoriented components. Is that always true? Can you sort the permutations of Exercise 2 with $d = n - c + k$ reversals? Can you do it with less, for item (c)?

TIP: for this exercise and the previous, the BP graph software from the WIKI page might be useful.

4. The number of possible (unsigned) permutations over $\{1, 2, \ldots, n\}$ is $n!$. Obviously, there exist bijective mappings between the numbers $1, 2, \ldots, n!$ and permutations over $\{1, 2, \ldots, n\}$. Find such a mapping that is computable in both directions in polynomial time. *Tip: Google is your friend.*